Lab Report

Lab Title: Spectrum Estimation of Multimedia Signals

Name: William Ewanchuk

Student ID: 1692036

Section: D51

Date Lab Performed: November 15, 2024

Question 1: Audio File Information

First Few Words of the Song:

I love to love, I love to love you,

So much I wanna share and do, wohohoh

I love to love, I love to love you,

I wanna find a way to you

(b) *MATLAB Code to Read the Audio File*

```
% Question 1: Load and Inspect the Audio File

% Load the audio file
[x, fs] = audioread('love_mono22.wav');

% Display the matrix size (number of samples) and sampling rate (fs)
disp('Size of the audio signal (samples x channels):');
disp(size(x));

disp('Sampling rate (Hz):');
disp(fs);

% Calculate the duration of the audio signal
duration = length(x) / fs; % Duration in seconds
disp('Duration of the audio signal (seconds):');
disp(duration);

% Assuming 8 bits per sample (as specified in the question)
bit_depth = 8;

% Calculate the bit-rate (in bits per second)
bit_rate = bit_depth * fs * length(x);
disp('Bit rate (bits/sec):');
disp(bit_rate);
sound(x,fs)
```

```
>> lab4question1
Size of the audio signal (samples x channels):
      400000           1

Sampling rate (Hz):
       22050

Duration of the audio signal (seconds):
   18.1406

Bit rate (bits/sec):
   7.0560e+10
```

Sampling Rate: Fs = 22050 Hz

Matrix Size of x: 400000 X 1 = size(x)

Bit-Rate Calculation: Bit-Rate = Bit-Depth × Sampling Rate

Bit-Rate (bits/sec): 7.0560 X 10^10 Bits/sec

Duration of Audio Signal (seconds): 18.1406

Question 2: Spectrum Estimation Using DFT

(a) MATLAB Code to Calculate the Discrete Fourier Transform

```
% Assuming x is the audio signal and Fs is the sampling frequency
[x, Fs] = audioread('love_mono22.wav');  % Read the audio file

% Calculate the FFT and scale
N = length(x);  % Number of samples
X = fft(x);  % Compute the DFT
X_prime = X/sqrt(N);  % Scale the coefficients

% Create the frequency vector
f = (Fs / N) * (0:N/2);

% Convert to dB scale
magX = 20* log10(abs(X_prime(1:N/2+1)))*2.3;

% Plot the magnitude spectrum
figure;
plot(f / 1000, magX);  % Frequency in kHz
xlabel('Frequency (kHz)');
ylabel('Magnitude (dB)');
title('Magnitude Spectrum');
```

```
Values of X[0], X[1], X[2]:
X[0] = -16.0625
X[1] = 0.77856-0.64057i
X[2] = 3.1697-1.6708i
```

(b) Values of X[0], X[1], and X[2]

X[0]: -16.0625

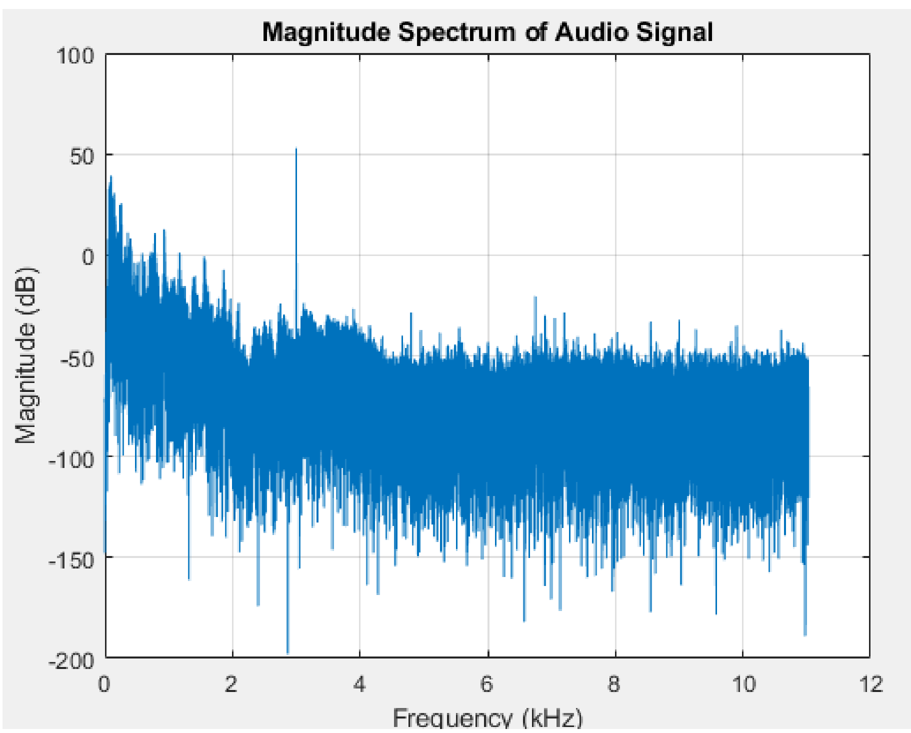
X[1]: 0.077856 -0.64057j

X[2]: 3.1697 -1.6708j

(c) MATLAB Code to Scale the Coefficients

```
% Load the audio file
[x, Fs] = audioread('love_mono22.wav');
N = length(x);  % Total number of samples

% (a) Calculate the Discrete Fourier Transform (DFT) of the audio signal
X = fft(x);  % Compute the FFT of the signal
frequencies = (0:N-1)*(Fs/N);  % Frequency axis for the DFT in Hz

% (b) Extract values of X[0], X[1], and X[2]
X0 = X(1);  % First coefficient
X1 = X(2);  % Second coefficient
X2 = X(3);  % Third coefficient
disp('Values of X[0], X[1], X[2]:');
disp(['X[0] = ', num2str(X0)]);
disp(['X[1] = ', num2str(X1)]);
disp(['X[2] = ', num2str(X2)]);

% (c) Scale the coefficients
X_scaled = X / sqrt(N);  % Scale the DFT coefficients

% (d) Plot the magnitude spectrum
magnitude_spectrum = abs(X_scaled);  % Magnitude of scaled DFT coefficients
magnitude_dB = 20 * log10(magnitude_spectrum) *2.3;  % Convert to decibels

% Scale frequency axis to KHz
frequencies_kHz = frequencies / 1000;

% Plot the magnitude spectrum in dB
figure;
% Plot positive frequencies
plot(frequencies_kHz(1:floor(N/2)+1), magnitude_dB(1:floor(N/2)+1));
xlabel('Frequency (kHz)');
ylabel('Magnitude (dB)');
title('Magnitude Spectrum of Audio Signal');
grid on;
```

(d) MATLAB Code to Plot the Magnitude Spectrum in dB



(e) Comments on the Magnitude Spectrum

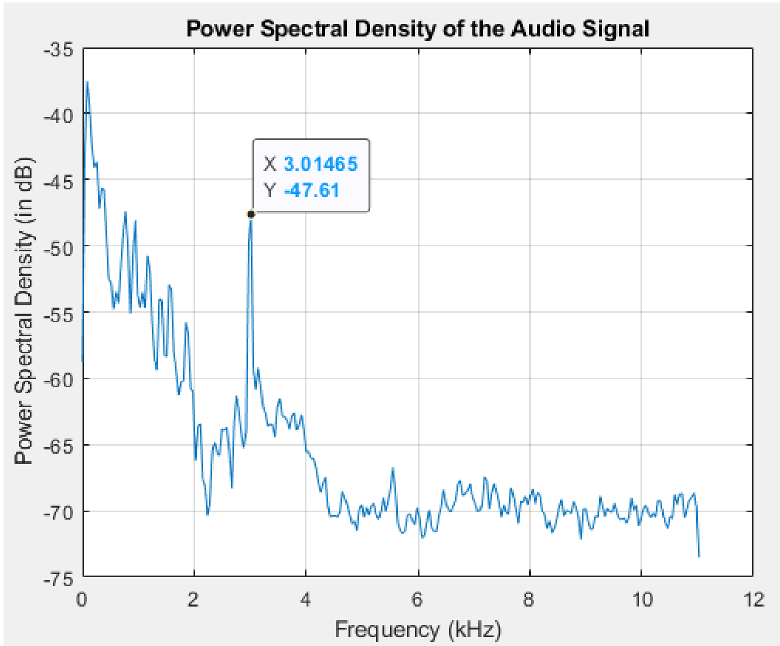
Comments:

- Most of the signal's energy is concentrated in the lower frequency range, particularly below 2 kHz. This indicates that the audio signal predominantly contains low-frequency components, which is typical of music or speech signals.
- The magnitude decreases significantly after approximately 2 kHz, showing a decline in the strength of higher frequency components. This is expected, as higher frequencies usually carry less energy in natural audio signals.
- A noticeable peak appears around 3, which most likely represents the tonal noise that could be heard throughout the sound clip, which was most likely the 3 kHz peak as there were no surrounding frequency peaks in the plot.

Question 3: Power Spectrum Using pwelch

(a) MATLAB Code to Generate the Power Spectral Density Plot

```
% Load the audio file
[x, Fs] = audioread('love_mono22.wav'); % Load the audio signal
N = 512;                                % FFT block size

% Estimate Power Spectral Density (PSD) using pwelch
[Px, F] = pwelch(x, N, [], N, Fs);

% Plot the PSD
figure;
plot(F/1000, 10*log10(Px)); % Frequency in kHz
xlabel('Frequency (kHz)');
ylabel('Power Spectral Density (in dB)');
title('Power Spectral Density of the Audio Signal');
grid on;

peak_power = max(10*log10(Px));          % Maximum power
threshold = peak_power - 20;            % Top 20 dB threshold
indices = find(10*log10(Px) >= threshold); % Indices within the range
freq_range = [F(indices(1)), F(indices(end))] / 1000; % Frequency kHz
disp(['Frequency range with most energy: ', num2str(freq_range(1)), ...
    ' kHz to ', num2str(freq_range(2)), ' kHz']);

[~, peak_indices] = findpeaks(10*log10(Px)); % Find peaks in the PSD
tonal_noise_frequency = F(peak_indices) / 1000; % Frequency in kHz
disp(['Tonal noise frequencies: ', num2str(tonal_noise_frequency')]);
```



(b) Frequency Range with Most Energy

Frequency Range with Top 20 dB Energy: 0.043066 kHz to 3.0146 kHz

- From the plot, the signal's energy is concentrated in the range 0–3 kHz.
- This is where the highest peaks occur, indicating that most of the power is within this frequency range.

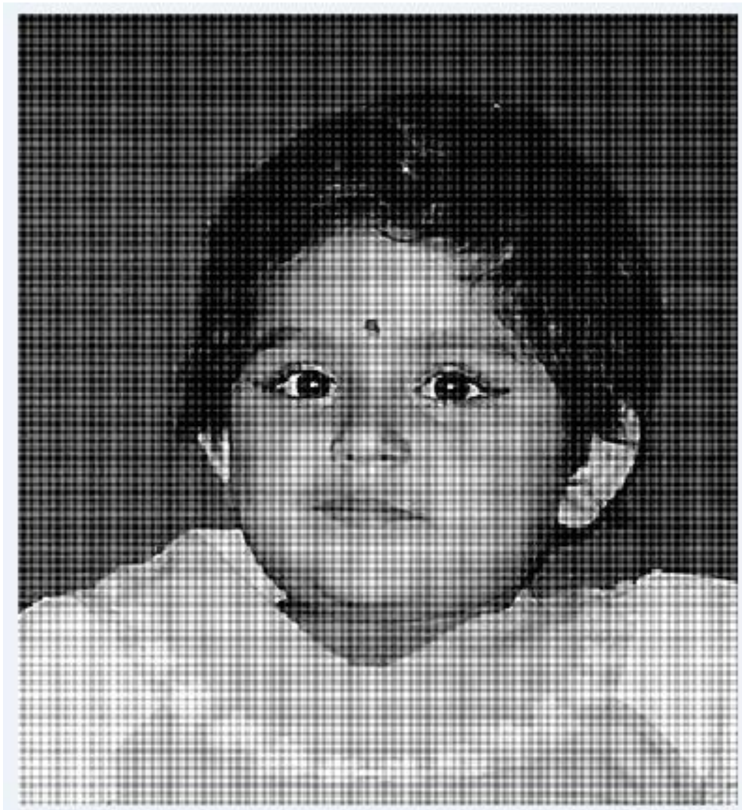
(c) Frequency of Tonal Noise

Frequency of Annoying Tonal Noise: 3.01465kHz

- From the plot, the signal's energy is concentrated in the range 0–3 kHz.
- This is where the highest peaks occur, indicating that most of the power is within this frequency range.

Question 4: Spectrum of a 2-D Image

Image with Visual Artifacts



(a) Comments on the Image Quality

Comments:

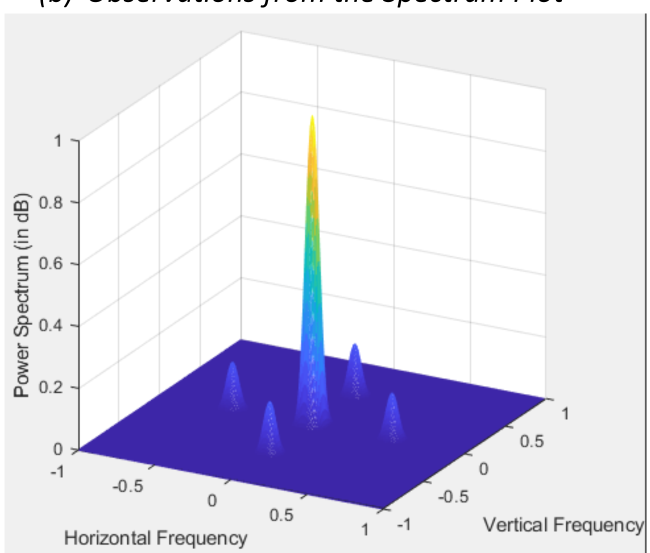
- The observed grid-like pattern in the image, resembling a 'picnic table' with gridded lines, appears to be an intentional distortion introduced into the signal. This pattern may correspond to a high-frequency modulation, or a deliberate **spatial filter** applied to the image. The periodic nature of the grid leads to repetitive structures in the image, which can be observed in both the visual domain and the frequency domain.

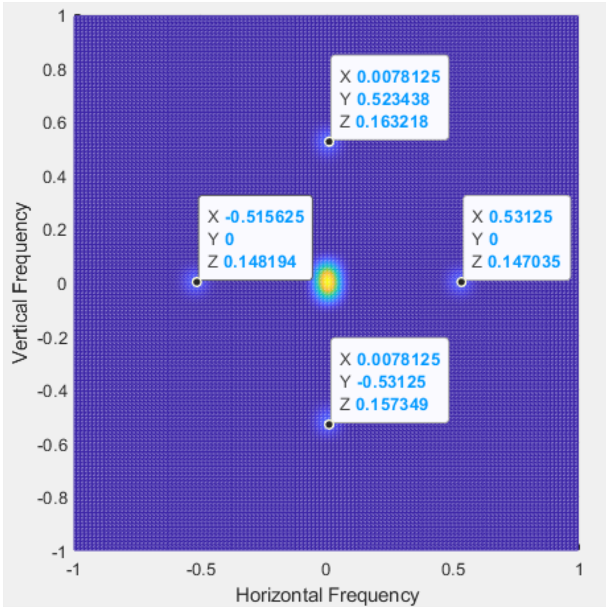
- The grid-like pattern observed in the image is a result of periodic noise introduced into the spatial domain. This periodicity corresponds to distinct high-frequency components in the Fourier spectrum. In the spatial domain, the grid lines appear as horizontal and vertical patterns that repeat across the image. In the frequency domain, these periodic structures manifest as distinct peaks at specific frequencies in the 2D spectrum. These peaks indicate the presence of energy concentrated at those spatial frequencies, directly caused by the repetitive grid pattern in the image.

MATLAB Code to Display and Analyze Image

```
clear
clc
% MATLAB code for spectral analysis and lowpass filtering of an image
% see section 17.6, Fig. 17.20, Fig. 17.21
%
% reading the image 'ayantika.tif'
I = imread('ayantika.tif');
imshow(I)                    % display the image
fprintf('\nThe image Ayantika has been displayed')
fprintf('\nPress any key to continue')
pause
I = double(I);
I = I - mean(mean(I));
% 2D Bartlett window
x = bartlett(32);
for i = 1:32
    zx(i,:) = x';
    zy(:,i) = x ;
end
bartlett2D = zx .* zy;
mesh(bartlett2D) ;
%
n = 0;
% calculate power spectrum
P = zeros(256,256);
for (i = 1:16:320)
    for (j = 1:16:288)
        Isub = I(i:i+31,j:j+31).*bartlett2D;
        P = P + fftshift(fft2(Isub,256,256));
        n = n + 1;
    end
end
Pabs = (abs(P)/n).^2;
mesh([-128:127]*2/256,[-128:127]*2/256,Pabs/max(max(Pabs)));
xlabel('Horizontal Frequency'); ylabel('Vertical Frequency');
zlabel('Power Spectrum (in dB)');
print -dtiff plot.tiff
```

(b) Observations from the Spectrum Plot





Normalized Frequency Ranges:

- Horizontal Frequency Range: [-1 , 1] cycles per pixel
- Vertical Frequency Range: [-1, 1] cycles per pixel

These ranges correspond to the spatial frequencies in the 2D Fourier domain after the Fourier Transform is performed, and the frequency resolution is determined by the size of the 256x256 pixellated grid. This is a normalized frequency scale, where 0 corresponds to the DC (low frequency) component, and the highest frequencies are represented by the edges of the plot.

(d) 2-D Frequencies of Noise Peaks

Noise Peaks Frequencies:

- [0.523438, 0]
- [-0.523438, 0]
- [0, 0.523438]
- [0, -0.523438]

The periodic grid pattern in the spatial domain directly corresponds to these frequencies, with the spacing between the grid lines determining the frequency values. Specifically, the closer the grid lines are to each other, the higher the spatial frequency. The symmetry of the frequency peaks (positive and negative) reflects the Fourier transform's properties, indicating that the same frequency content exists in both directions visually.

Discrete- Time Signals and Systems

Lab 4 Marking Sheet

Name: William Ewanchuk Student ID: 1692036

Lab Section: D51 Lecture Section: A2

	Submitted	Score
Q1(b)	✓ Code to read the audio file? ✓ Sampling rate? ✓ Calculate bit-rate and duration?	/3
Q2(a)	✓ Code to calculate DFT?	/2
Q2(b)	✓ Value of $X[0], X[1]$ and $X[2]$?	/2
Q2(c)	✓ Code for scaling $X[r]$?	/1
Q2(d)	✓ Code to generate magnitude plot (in dB) and plot? ✓ Code for proper scaling of frequency axis?	/2
Q2(e)	✓ Comments?	/1
Q3(a)	✓ Generated pwelch plot?	/1
Q3(b)	✓ Frequency range in which the signal has most energy?	/2
Q3(c)	✓ Frequency of the tonal noise?	/1
Q4(b)	✓ Comments and observations about the image?	/2
Q4(c)	✓ Generated Spectrum of the image	/1
Q4(d)	✓ 2-D frequencies of the noise peaks?	/2
Report format and documentation		/5
Total		/25